

Developing and Deploying Custom Assemblies with SQL Server Reporting Services 2000

Excerpted from a workshop I taught at
Microsoft ASP.NET Connections
Orlando, Spring 2005

Paul Litwin

- **Developer**
 - Focus: ASP.NET, ASP, VB, C#, SQL Server, ...
 - MCSD
 - Microsoft MVP
 - Lead Programmer with Fred Hutchinson Cancer Research Center
- **Co-Founder and Senior Trainer**
 - Deep Training
 - www.deeptraining.com
- **Conference Chair/Speaker**
 - Chair, Microsoft ASP.NET Connections
 - Member INETA Speakers Bureau
- **Author**
 - Author/co-author of a dozen books, including...
 - *ASP.NET for Developers*
 - *Access Cookbook, 2nd edition*
 - *Access 2002 Desktop/Enterprise Dev Handbook*

Samples Download

- Download samples from
 - www.deeptraining.com/litwin/

Developing & Deploying Custom Assemblies

NOTE:

This presentation applies to SSRS 2000

It has not been tested/verified if it works the
same way in SSRS 2005

Custom Assemblies

- Why not just use Code window?
 - Reusability
 - Desire to code the solution in C# (or the *real* VB.NET)
 - Ability to do things that you can't do within the code window (e.g., database access, etc.)
- Challenges
 - Reference needed
 - Security issues

Problem

- At FHCRC, we had created a complex report built with many overlapping sections
- Report was to replace convoluted solution based on Excel and Access
- Excel allowed the users to use denominators in percent calculations from anywhere else in the report (random access percentages)
- SRS, however, only allows you to calculate percentages based on the group total or grand total

Solution: Custom Assembly

- We created a custom assembly that allowed us to store away a value into a static hashtable and later retrieve the value back out for calculating percentages
- Complicating security issue: the assembly needed access the database

Steps to Follow When Developing Custom Assembly (1 of 2)

1. Create assembly
2. Test assembly using ASP.NET, WinForm, or Console app
3. Copy assembly to two folders on report server
 - a. C:\Program Files\Microsoft SQL Server\80\Tools\Report Designer
 - b. C:\Program Files\Microsoft SQL Server\MSSQL\Reporting Services\ReportServer\bin
4. Reference assembly from report
5. Call methods using syntax:
=project.class.method()
6. Run report in preview mode (if you need to change source code you will need to exit and restart VS.NET)

Steps to Follow When Developing Custom Assembly (2 of 2)

7. Time to address security issues (*plan to spend some time on this step*)
 - a. Make assertion of permission needs within code
 - b. Modify code access security settings within two SRS policy files
 - i. C:\Program Files\Microsoft SQL Server\80\Tools\Report Designer\rspreivpolicy.config, and
 - ii. C:\Program Files\Microsoft SQL Server\MSSQL\Reporting Services\ReportServer\rssrvpolicy.config
 - c. See <http://support.microsoft.com/?kbid=842419>, Chapter 9 of Blackburn, Vaughn book, and <http://blogs.sqlxml.org/bryantlikes/articles/824.aspx> (blog post by Bryant Like) for more details
8. Deploy and run report on server

RSSupportCS Example

- Original solution was built in VB
- Version of solution presented here built in C#
- RSSupportCS class library project
 - RASupport class ?RAPercent class
 - Save method •GetPercent method
 - Get method

RSSupportCS Example

- Asserting permission needs within custom assembly

```
public static void Populate() {
    System.Data.SqlClient.SqlClientPermission scp;
    scp = new SqlClientPermission(
        System.Security.Permissions.PermissionState.Unrestricted);
    scp.Assert();

    string strCnx = "Data Source=localhost;...";
    SqlDataAdapter sdaRAPctInfo = new SqlDataAdapter("procRAGetPctInfo",strCnx);
    sdaRAPctInfo.SelectCommand.CommandType = CommandType.StoredProcedure;

    dtRAPctInfo = new DataTable();
    sdaRAPctInfo.Fill(dtRAPctInfo);
    //...continues ...
}
```

RSSupportCS Example

- Modifying code access security settings in RS policy files (use Microsoft .NET Framework 1.1 Configuration tool to help you build as noted in KB #842419 article)

```
<PermissionSet class="NamedPermissionSet"
version="1"
Name="RunRSCustomCode"
Description="Execute code within RSSupportCS assembly...">
<IPermission class="SecurityPermission" version="1" Flags="Assertion, Execution"/>
<IPermission class="SqlClientPermission" version="1" Unrestricted="true"/>
</PermissionSet>

<CodeGroup class="UnionCodeGroup"
version="1"
PermissionSetName="RunRSCustomCode"
Name="RSCustomCode"
Description="Allows execution of RSSupportCS custom assembly from ...">
<IMembershipCondition class="UrlMembershipCondition"
version="1"
Url="file://C:/Program Files/Microsoft SQL Server/80/Tools/Report Designer/RSSupportCS.dll"/>
</CodeGroup>
```

Additions to config Files

- Location, location, and location!

```
<NamedPermissionSets>
...
  <PermissionSet
  </PermissionSet>
  [---Insert new PermissionSet element here ---]
</NamedPermissionSets>

<CodeGroup
...
  [---Insert new CodeGroup element here ---]
  </CodeGroup>
</CodeGroup>
```